Applied Network Science

RESEARCH Open Access

Analyzing and inferring human real-life behavior through online social networks with social influence deep learning

Luca Luceri[1,2*], Torsten Braun[2] and Silvia Giordano[1]

*Correspondence: luca.luceri@supsi.ch
[1]University of Applied Sciences and Arts of Southern Switzerland (SUPSI), Manno, Switzerland
[2]University of Bern, Bern, Switzerland

Abstract

The advent of Online Social Networks (OSNs) has offered the opportunity to study the dynamics of information spread and influence propagation at a huge scale. Considerable research has focused on the social influence phenomenon and its impact on OSNs. Social influence plays a crucial role in shaping people behavior and affecting human decisions in various domains.

In this paper, we study the impact of social influence on offline dynamics to study human real-life behavior. We introduce Social Influence Deep Learning (SIDL), a framework that combines deep learning with network science for modeling social influence and predicting human behavior on real-world activities, such as attending an event or visiting a location. We propose different approaches at varying degree of network connectivity with the objective of facing two typical challenges of deep learning: interpretability and scalability.

We validate and evaluate our approaches using data from Plancast, an Event-Based Social Network, and Foursquare, a Location-Based Social Network. Finally, we explore the usage of different deep learning architectures, and we discuss the correlation between social influence and users privacy presenting results and some notes of caution about the risks of sharing sensitive data.

Keywords: Social influence, Information diffusion, Deep learning

Introduction

Information diffusion in techno-social systems has received tremendous interest in the last decade. The advent of Online Social Networks (OSNs), and their intrinsic multi-relational data, has offered the opportunity to study the dynamics of information spread and influence propagation at a huge scale. Considerable research has focused on the diffusion of information in OSNs, which is also referred to as electronic Word of Mouth (eWOM) (Brown et al. 2007; Goldenberg et al. 2001; Huete-Alcocer 2017). The powerful influence of eWOM can shape individuals' actions, affect their decisions, and also be exploited to manipulate their opinions. On one hand, OSNs have been shown to be abused for nefarious purposes (Ferrara 2015; Bessi et al. 2015; Luceri et al. 2019), such as astroturf campaigns (Metaxas and Mustafaraj 2012; Ratkiewicz et al. 2011), antivaccination movements (Tangherlini et al. 2016; Subrahmanian et al. 2016), and stock market manipulation (Hwang et al. 2012; Ferrara et al. 2016). On the other hand, the understanding of

Springer Open

© The Author(s). 2019 **Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

how influence propagates in OSNs opens the door to a wide range of applications more beneficial for users, such as targeted advertising, viral marketing, and recommendation. This has become possible as OSNs do not only connect people, by providing a medium for spreading processes (Newman 2003; Albert and Barabási 2002; Castellano et al. 2009), but also (and most importantly) reveal preferences, activities, and interests of their users over the time.

For these reasons, influence propagation received huge interest, both in academia and industry, providing multiple applications in various fields. In particular, many efforts have been devoted to the comprehension and modeling of the social influence phenomenon. Social influence is recognized as a key factor that governs human behavior and drives individual decisions. The main idea behind social influence is that the interaction with other individuals (or a group) may affect or change subjects' thoughts, feelings, or behavior. Though social interactions may occur both online (through OSNs) and offline, social influence underlies real life spreading phenomena, such as the diffusion of opinions and the adoption of products, with inevitable repercussion on marketing, politics, health, and business (Domingos and Richardson 2001; Mønsted et al. 2017). A considerable amount of work has been conducted to investigate social influence and analyze its effects (Aral et al. 2009; Bakshy et al. 2011; La Fond and Neville 2010). In Singla and Richardson (2008) and (Anagnostopoulos et al. 2008), the authors propose how to qualitatively measure the existence of social influence, whereas in Crandall et al. (2008) the correlation between social similarity and influence is examined. In Luceri et al. (2017); Luceri (2016), we introduce a novel interpretation of physical, homophily, and social community, as sources of social influence. Along with these qualitative studies, complementary research efforts have focused on developing predictive models of diffusion processes (Guille et al. 2013). This broad research area presents two classes of social influence modeling. We can distinguish macro- and micro-level models according to the outcome granularity. While the former class (Matsubara et al. 2012; Myers et al. 2012) focuses on predicting the result of a diffusion process at the network level (e.g., number of adopters, spreaders, or infected individuals overall the network), the latter aims to study social influence at the user-level, providing prediction of a given spreading process for each subject (Goyal et al. 2010; Saito et al. 2008).

Micro-level social influence modeling

In this paper, we focus on social influence at the user-level as we are interested in measuring whether and to what extent an individual is influenced by other subjects. More specifically, we leverage on dyadic social interactions between subjects to predict their behavior. Such micro-level analysis suits several real-life applications, such as targeted advertising, recommendation, and viral marketing. In particular, viral marketing is a convenient example of how to exploit social influence to maximize the adoption of a product or, more in general, the information spread. Although influence maximization (Domingos and Richardson 2001; Richardson and Domingos 2002; Kempe et al. 2003; Kimura et al. 2007) is not the target of this paper, the seminal models presented in Kempe et al. (2003) provide the underpinning of multiple existing approaches to model diffusion processes in social networks. The models proposed by Kempe et al. (2003), referred to as Independent Cascade (IC) model and Linear Threshold (LT) model, map a spreading process to every single node of an underlying graph. In the IC model, each subject independently influences her friends with given influence probabilities (the power to influence neighbors (Goyal et al. 2010)), while in the LT model, a subject is influenced by her friends if the combination of their total influence probabilities exceeds a threshold. Both models assume to have as input a social network whose edges are weighted by a measure of influence probability, as shown in the example in Fig. 1. However, these values are not known in practice and, thus, they need to be estimated.

Many efforts have been made to quantitatively measure the influence probability between pairs of friends (Gruhl et al. 2004; Saito et al. 2008; Tang et al. 2009; Goyal et al. 2010; Liu et al. 2012; Fang et al. 2013). Existing works explored different forms of social influence. In Tang et al. (2009), the authors proposed a graphical probabilistic model to measure influence strength. In their approach, they analyze influence propagation at the topic-level with the objective of learning influence probabilities with respect to given topics. Similarly, Gruhl et al. (2004) characterize information diffusion by tracking topics and individuals across different blogs. On the other hand, other approaches (Goyal et al. 2010; Saito et al. 2008) offer more general models (topic and domain independent) by leveraging only the history of the actions performed by each subject. In particular, Goyal et al. (2010) rely on an instance of the LT model introducing different metrics to estimate the pairwise influence between two individuals. In Saito et al. (2008), authors focused on the IC model and employ the Expectation-Maximization (EM) algorithm to estimate the influence probability associated with each edge. More recently, Zhang et al. (2013, 2015) studied social influence on Twitter and introduced the concept of social influence locality on users' retweet behavior. They proposed two approaches, i.e., a logistic regression classifier and a factor graph model, based on social influence locality and three kinds of hand-crafted features.

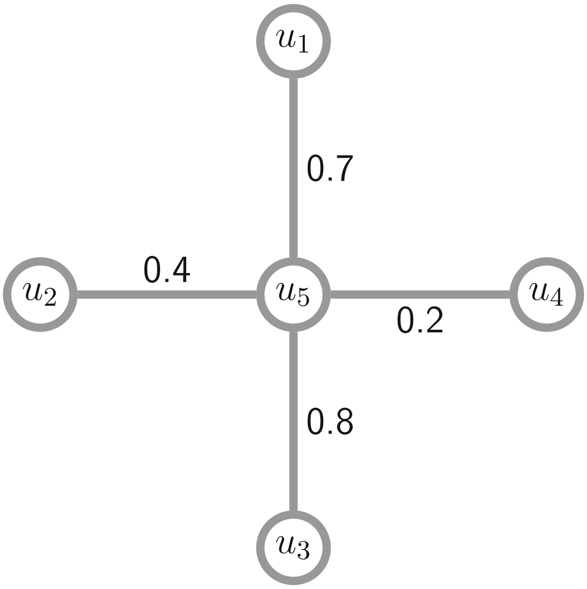


Fig. 1 Ego network of user u_5. Each node connected to u_5 represents a friend, while the weight on the edge indicates the influence probability

Contributions

In this work, we propose a social influence model for forecasting influence propagation in real-life scenarios. Although our model can be generalized to every kind of human activity (both online and offline), we focus on real-world (offline) activities, such as attending an event or visiting a location. To the best of our knowledge, this is the first work that seeks to provide a social influence model of real-life activities. Similar works focused on online activities, such as following, grouping, voting, tagging, etc., in OSNs and blogs. For this purpose, we analyze data from Plancast, an Event-Based Social Network (EBSN), and Foursquare, a Location-Based Social Network (LBSN). Such platforms provide remarkable opportunities to analyze users' real-world behavior and interactions from the lens of online social media.

Similarly to (Goyal et al. 2010; Saito et al. 2008), the proposed approach does not require any specific knowledge of the domain under analysis and can be applied to a variety of contexts. In fact, we aim at learning influence strengths among subjects by leveraging the actions performed by users in their history and how such actions propagated between each other. Thereby, our model takes as input only the *raw data* related to users' actions in OSNs (i.e., the action log) and, thus, does not require any hand-crafted features, which in turn may depend on the specific OSN and on the availability of metadata. Although the models suggested in Goyal et al. (2010); Saito et al. (2008) can be generalized to various contexts, they have two main drawbacks: (*i*) they assume that the probability of friends influencing a subject are independent of each other, and (*ii*) they do not consider the actions not performed by the subject (but performed by her friends) to learn the influence probabilities.

In our preliminary study (Luceri et al. 2018), we showed that these two limitations can be overcome by employing a deep learning approach. We proposed the usage of Deep Neural Networks (DNNs) to model social relationships embedded in OSNs and learn the interplay among their users. This work represents the natural completion of our previous intuition. In particular, here we expand our previous approach in a framework, called Social Influence Deep Learning (SIDL), for modeling and forecasting social influence. We propose different SIDL approaches at varying degree of network connectivity ranging from the *local* ego network of each user to the totality of the social network. Further, we extend our experiments using data from two different OSNs (an LBSN and an EBSN) to validate and evaluate the proposed approaches. Moreover, we explore the usage of different deep learning architectures, and we discuss the relationship between social influence and users' privacy presenting results and some notes of caution about the risks of sharing sensitive data. Finally, we address two main typical challenges of deep (machine) learning models: interpretability and scalability.

Although DNNs have reached outstanding performance on different tasks, they produce an obscure model and, for this reason, they are referred to as black boxes. A black box is defined as a predictor, whose internals are either unknown to the observer or they are known but uninterpretable by humans (Guidotti et al. 2018). Contrarily, an interpretable solution is desirable as deep learning is nowadays used in critical areas, such as justice and medicine, where the understanding of the model logic, functionality, and results can be necessary. However, according to the analysis provided by Lipton (2016), interpretability holds no agreed-upon meaning, despite numerous papers assert the interpretability of their models. In this study, the author claims that interpretability is not a monolithic concept, but it reflects several distinct ideas. For this reason, to be meaningful, any assertion regarding interpretability should fix a specific definition. Lipton describes two categories of interpretability. The first is related to the concept of transparency (as the opposite of blackbox-ness) and focuses on the understanding of the model, its components, and training algorithms. The second category, namely *post-hoc explanations*, does not aim to explain how a model works but has the objective of extracting information from a learned model. In particular, given the output of a black box predictor, the problem consists in reconstructing an explanation for it, without necessarily elucidate the logic behind the model. This problem is called *outcome explanation problem*, and it is the notion of interpretability we study in this work. One of the most common technique to solve the outcome explanation problem is to focus on explaining what a neural network depends on *locally*, instead of trying to understand the full mapping learned by the model. This *local explanation* aims to predict the response of the predictor in a neighborhood of a given input. One of the technique generally used to accomplish this purpose is Saliency Mask (SM). SM aims to explain the DNN outcome by identifying a subset of the input, which is mainly responsible for the prediction. As an example, in Fong and Vedaldi (2017), SM is used to highlight the salient part of the images that causes a certain outcome.

While interpretability is a more rational problem, scalability represents a practical issue related to DNN implementation. The idea is that the social influence model should be able to adapt and scale to previously unseen users. In this sense, scalability issues may occur when new users register to a social network. A new user can perform activities, create social connections with other OSN users, and influence them. In such a case, the social influence model should be adjusted in order to consider new users in the social network. Our objective is to mitigate this issue in the most efficient way, in terms of time and resource consumption, while preserving performance.

Towards meeting these challenges, in this paper, we propose to combine network science with deep learning. We consider different models by varying the network structure in which the user in embedded. We first describe the Global-SIDL approach, which considers the whole social network in a unique model. We then narrow our approach to the ego network of each individual and build a *local* model (Local-SIDL) for each user in the dataset. The idea is to follow the previously described principle of local explanation by considering the neighborhood (friends) of a given input (user) to explain the output (user behavior). Finally, we propose to decompose the social network in mesoscale level structures, here identified through a community detection algorithm, so as to consider a larger social structure within each user is embedded, while maintaining a good trade-off between performance, interpretability, and scalability.

Problem definition

Let $G = (V, E)$ be an undirected graph representing the social network, where $V = \{u_1, u_2, \dots, u_N\}$ is the set of users, and E is the set of edges that connect them. The edge $(u_i, u_j) \in E$ indicates a social tie between u_i and u_j, which in turn are referred to as *friends*. We denote with $\mathcal{A}_l$ the *action log*: a record of the actions performed by every user in the social network. Each entry of the action log $\mathcal{A}_l$ is a tuple (t_i, u_i, a) representing the action a performed by user $u_i \in V$ at time t_i. Let A be the set of the actions performed in $\mathcal{A}_l$, for each action $a \in A$, each user is either active (if she performed the action) or inactive (otherwise). In accordance with (Goyal et al. 2010; Saito et al. 2008), we say that an action $a \in A$ propagates from u_i to u_j if the two following conditions are met:

- $(u_i, u_j) \in E$,
- $(t_i, u_i, a), (t_j, u_j, a) \in \mathcal{A}_l$ with $t_i \leq t_j$.

We consider the scenario where $t_i = t_j$ to take into account the mutual influence between users in performing an activity at a fixed time $t = t_i = t_j$, e.g., attending an event. Finally, we indicate with A_{u_i} the set of actions performed by u_i and with S_{a,u_i} the set of active friends of u_i for the action a.

In this paper, we keep track of user activities over time to model complex social relationships among them. In particular, we focus on the comprehension and modeling of the social influence phenomenon, with the final objective of forecasting influence propagation in real-world scenarios. The idea behind our social influence modeling is to learn influence strengths among subjects by leveraging the actions performed by users in their history and how such actions propagated between each other. Once the model is trained, we target to exploit it to predict users' real-life behavior as a consequence of other users' actions. More specifically, our objective is to infer whether users will perform action a based on their active friends $S_{a,u_i}, \forall a \in A$. The rationale of this approach is based on the concept of social influence itself. A subject may perform an action, e.g., to buy a new product or to watch a TV show, when her friends have performed the same action.

To this aim, we introduce Social Influence Deep Learning (SIDL), a deep learning framework for both modeling and forecasting social influence. SIDL is based on Deep Neural Networks (DNNs), a class of machine learning algorithms inspired by biological nervous systems. The rationale of SIDL is the capability of a DNN to automatically extract complex relationships embedded in the input data by means of its multi-layer architecture (He et al. 2017). Thereby, if we represent each user in the social network as an input node of the DNN we can model the interplay among users through the DNN layers. For each user, we consider the history of the actions propagated from her friends to train the DNN and tune the model parameters. Once the DNN is trained, we utilize the influence model to predict new (not performed yet) activities based only on the activities performed by the user's friends.

SIDL includes different approaches at varying degree of network connectivity with the purpose of finding both a scalable and interpretable solution. Thereby, we distinguish: (*i*) a network-based model, called Global-SIDL (G-SIDL), which takes into account the whole social network; (*ii*) a user-based model, called Local-SIDL (L-SIDL), which considers only the ego network of the user; and finally (*iii*) a community-based model, referred to as Community-SIDL (C-SIDL), which decomposes G-SIDL in smaller sub-models based on social network partitions.

Data

For validating and evaluating the SIDL framework, we consider different datasets from two OSNs. In particular, we focus on scenarios of real-life activities, such as attending an event or visiting a location. The idea is that a subject might participate in an event because she sees her friends taking that decision, or she may visit a location (e.g., a bar or a restaurant) because some friends have been there before and suggested her that venue. For this reason, we analyze data from Plancast, an Event-Based Social Network (EBSN), and Foursquare, a Location-Based Social Network (LBSN). EBSN and LBSN provide remarkable opportunities to analyze users' real-world behavior through OSNs. In these scenarios, the set of actions A is defined by the event participation and location visits in the EBSN and LBSN, respectively. Both of these actions reveal the interactions between users and the real world (Esfandyari et al. 2016), which is widely different from the virtual world and its online activities (following, grouping, voting, tagging, etc.) (Gao et al. 2012). Thereby, $A_{u_i} \subseteq A$ represents the set of events (locations) attended (visited) by subject $u_i \in V$, while a subject is considered active for the action a if she decided to participate in (visit) the event (location) $a \in A$.

Table 1 and Fig. 2 summarize properties and statistics of both the OSNs analyzed in this study. It should be noticed that we do not model users across the two OSNs as their IDs have been anonymized before the data release. Thus, it is not possible to match users from different datasets.

Foursquare

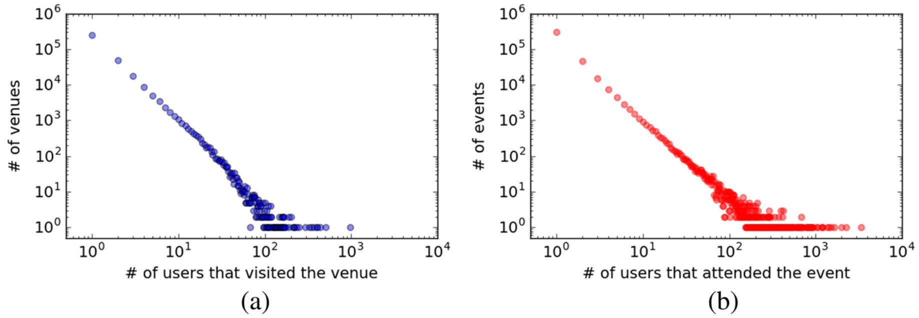
In recent years, LBSNs have become popular services that allow users to register (*check-in*) at named places and share their location with their friends. Check-in information includes latitude, longitude, category, the ID of the location, and time of the check-in. Historical check-ins provide useful hints about user interests and preferences, and represent a promising source of activity data to study human behavior and social dynamics (Noulas et al. 2011).

In this study, we analyze data from Foursquare, one of the most popular LBSN. We explore a dataset collected in Bao et al. (2012), which gathers Foursquare check-ins from the cities of New York and Los Angeles. The first dataset (New York) was collected for 30 months from May 5, 2008, while the second dataset (Los Angeles) gathered check-ins for 36 months from February 5, 2009. In Foursquare, users can check-in their location through a mobile application, give recommendations (*tips*), connect with their friends, and share with them their experiences. In such a scenario, users have the chance to discover new venues, look for trend places, and read friends' reviews. This bundle of information can produce a social contagion effect, which may affect user activities.

Table 1 Basic statistics of the OSNs used in this study

	Foursquare		Plancast
	NYC	LA	
n	47240	30207	75598
m	596379	246560	1501618
z	25.25	16.32	39.7
l	10	12	11
C	0.14	0.15	0.23
D	0.0005	0.0005	0.0005
p	425692	268102	869200
v	206098	144348	401634

The properties measured are: total number of nodes $n = |V|$; total number of edges $m = |E|$; average degree z; diameter l; average clustering coefficient C; network density D; total number of performed actions $p = |\mathcal{A}_l|$, i.e., attending an event or visiting a location; total number of venues v, i.e., events or locations



(a)

(b)

Fig. 2 Aggregate Statistics for the Foursquare (on the left) and Plancast (on the right) dataset: (**a**) and (**b**) depict the number of users that visited (attended) a venue (event)

Plancast

An EBSN is a web platform where users can create events, promote them, and invite friends to participate. Events range from small get-together activities, e.g., Sunday brunch or movie night, to bigger events, e.g., concerts or conferences (Liu et al. 2012). The rationale behind the choice of utilizing an EBSN is the intrinsic agglomerative power of the events. In fact, participation in an event represents a direct and explicit form of social interaction, other than a personal interest. An EBSN provides a social network service to connect friends and users with common interests. In the event main page, a user can see the information related to the event, e.g., date, location, and description, along with the confirmed participants. This information may activate processes of social influence, which can drive user participation in the events (Georgiev et al. 2014; Luceri et al. 2017).

In this study, we use a dataset collected by Liu et al. (2012) from the EBSN *Plancast* for three months (from September to November 2011). Plancast allows users to subscribe to each other providing direct connections among them. Subscription is similar to the concept of *following* on Twitter. Users can directly follow friends' event calendars: this mechanism allows a subject to be aware of friends' interests, event creation, and participation.

Methodology

In this Section, we first provide a short introduction to DNNs to motivate and introduce the SIDL framework. We then present the SIDL approaches and corresponding implementations.

Deep neural networks

Deep Learning is a fancy marketing name for artificial neural networks, a class of machine learning algorithms inspired by biological nervous systems. In recent years, neural networks (LeCun et al. 2015; Schmidhuber 2015) have found successful applications in a growing number of areas ranging from speech and image recognition to natural language processing and computer vision. This is confirmed by the intensive research and development that has been carried out during the last years in numerous fields, not only directly related to Artificial Intelligence (AI) applications.

An artificial neural network is defined by a combination of three layers: input layer ($\mathbf{x}$), hidden layers ($\mathbf{h}$), and output layer ($\mathbf{y}$). A Deep Neural Network (DNN) is an artificial neural network with multiple hidden layers ($\mathbf{h} = \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L$) between the input and output layers. Each layer is composed of multiple processing units (*neurons*), which use the output from the previous layer as input. The cascade of multiple layers is connected as

$$\mathbf{h}_j = \begin{cases} \phi_j(\mathbf{x}\mathbf{W}_{xh_j}) & \text{if } j = 1 \\ \phi_j(\mathbf{h}_{j-1}\mathbf{W}_{h_{j-1}h_j}) & \text{if } 1 < j \leq L \end{cases}$$

$$\mathbf{y} = \phi_o(\mathbf{h}_L \mathbf{W}_{h_L y}) \ ,$$

where $\mathbf{W}_{kl}$ indicates the weights of the connections between layer k and l, while ϕ_j is a non-linear activation function (e.g., sigmoid, ReLU, tanh, softmax) of each hidden node at layer j, and ϕ_o is a non-linear activation function of each output node.

The cascade of multiple layers consisting of non-linear neurons allows a DNN to approximate any continuous function. Moreover, DNN replaces the manual feature extraction procedure by building up a complex hierarchy of concepts (abstractive features) through the multiple layers to automatically extract relationships embedded in the input data (He et al. 2017). The predictive model of a DNN can be formulated as $\hat{\mathbf{y}} = f(\mathbf{x}|\Theta)$, where $\hat{\mathbf{y}}$ denotes the predicted output, Θ represents the model parameters (i.e., the inter-layers weights), and f indicates the function that maps the input $\mathbf{x}$ to the output $\hat{\mathbf{y}}$ based on the DNN architecture, i.e.,$f(\mathbf{x}) = \phi_o(\phi_L(\ldots \phi_2(\phi_1(\mathbf{x}))\ldots))$.

Social Influence Deep Learning (SIDL)

SIDL is a DNN-based framework for both modeling social influence and predicting user behavior. SIDL has the capability to accomplish these two goals in one shot (Luceri et al. 2018). The rationale of this solution is based on the capability of a DNN to extract relationships embedded in the input data. Thereby, if we represent each user in the social network as an input node of the DNN, we can model the interplay and dependencies among users by leveraging their activity history.

To learn social influence among users, we make use of a *training set* extracted from the action log $\mathcal{A}_l$. For each user u_i (from now on *target-user*), we consider the history of the actions propagated from her friends to the target-user to train the DNN and tune the model parameters Θ. Once the DNN is trained, we utilize the influence model to predict the target-user's new (not performed yet) activities according to her friends' activities. This task can be formulated as the problem of predicting whether subject u_i will perform action a as a function of her active friends S_{a,u_i}. We consider this task as a binary classification problem, where the DNN output $y_{u_i,a}$ is a Boolean variable that is equal to 1 if the target-user u_i performed a, and is 0 otherwise.

Global-SIDL (G-SIDL)

The first SIDL approach we present is based on the idea of modeling the entire social network in a unique neural network, thus, we name this solution *Global-SIDL* (G-SIDL). The rationale of G-SIDL is to have a unique model that includes every user in the social network to learn the interplay among individuals and model their inter-dependencies. To accomplish this purpose, we employ a DNN structured as follows: The input layer is composed of two concatenated vectors referred to as *target-user ID vector* $\left(\mathbf{v}_{u_i}^{ID}\right)$ and *social network vector* $\left(\mathbf{v}_{u_i}^{SN}\right)$, respectively. Both of them have length $N = |V|$. The former is a one-hot vector that uniquely identifies each target-user $u_i \in V$. One-hot encoding is widely used in machine learning to distinguish the elements of a set. The target-user

ID vector consists of all *zeros* with the exception of a single *one* that identifies the target-user, e.g., u_i is represented by the vector $\mathbf{v}_{u_i}^{ID}$, whose i^{th} element is the only element equal to one. The latter gives a representation both of the social network connections and of the users' activity. In particular, each element represents a user in the social network and its value indicates the state (active/inactive) of the user for a given action a. Thereby, the social network vector $\mathbf{v}_{u_i}^{SN}$ of the target-user u_i represents the social network of u_i and the state of her active friends. The j-th element of $\mathbf{v}_{u_i}^{SN}$ corresponds to user u_j and the corresponding input value is computed as follows:

$$\mathbf{v}_{u_i}^{F_a}(j) = \begin{cases} 1 \text{ if } (u_j, u_i) \in E \text{ and } u_j \text{ is active} \\ 0 \text{ otherwise} \end{cases}$$

These two vectors are first concatenated and then fed into a multi-layer architecture, as depicted in Fig. 3, where, for the sake of simplicity, a DNN with only one hidden layer is depicted. In our experiments, we design a network with a tower structure, where the bottom layer is the largest and the number of nodes of each successive layer is half of its precedent. In such a way, according to He et al. (2016, 2017), higher layers with few nodes can learn more abstractive features from the input data. Details about the implementation will be given in "Data processing and implementation" section. The output of the DNN $y_{u_i,a}$ corresponds to the target-user u_i and is equal to 1 if she performed a, and is 0 otherwise. The predictive model of the G-SIDL can be formulated as $\hat{y}_{u_i,a} = f\left(\mathbf{v}_{u_i}^{ID}, \mathbf{v}_{u_i}^{SN} | \Theta\right)$, and the training is performed by minimizing the cost function

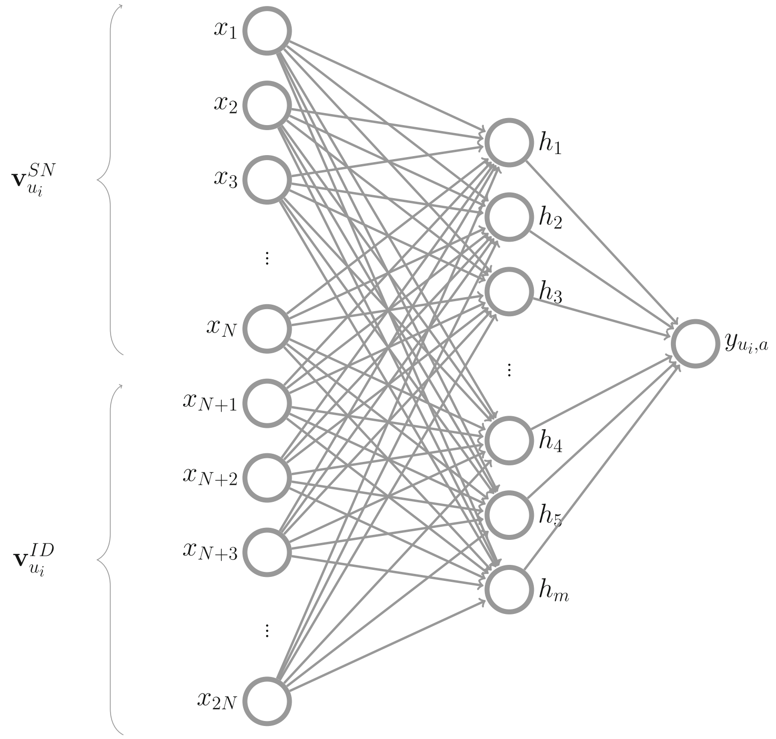


Fig. 3 Global-SIDL (G-SIDL). The input layer is composed of two concatenated vectors referred to as *target-user ID vector* ($\mathbf{v}_{u_i}^{ID}$) and *social network vector* ($\mathbf{v}_{u_i}^{SN}$), respectively. The output $y_{u_i,a}$ corresponds to target-user u_i, and is equal to 1 if she performed a, and is 0 otherwise

$\mathcal{L} = -y_{u_i,a} \log(\hat{y}_{u_i,a}) - (1 - y_{u_i,a}) \log(1 - \hat{y}_{u_i,a})$, which is referred to as binary cross-entropy loss.

Although the global approach achieves promising results (Luceri et al. 2018), this architecture may present two issues: interpretability and scalability. Note that, as we clarified in the introduction, the notion of interpretability used throughout the paper is related to a post-hoc explanation of the results. More specifically, we aim at understanding which subset of the input data is mainly responsible for the prediction outcome (Guidotti et al. 2018). As each input of G-SIDL represents a social network user, the global model does not provide a comprehensible explanation of the results. In fact, G-SIDL maps the interplay between all the users and does not allow to identify the subset of input responsible for the prediction related to a given target-user, i.e., we cannot identify a subset of individuals that mainly influence the target-user.

Towards meeting this challenge, we exploit the notion of local explanations and, more specifically, the principle behind the Saliency Mask (SM) technique. SM is defined as a summarized explanation of where the classifier "looks" to make its prediction (Dabkowski and Gal 2017). A suitable example is provided by the image classification scenario, where SM is used to find the part of an image most responsible for the classifier decision (Fong and Vedaldi 2017). The idea behind SM is to understand what a neural network depends on *locally*, i.e., to identify a subset of the input used by the model to produce the output.

In this paper, we reinterpret the usage of such a technique in the context of social networks. According to the concept of local explanation, here we narrow the input space by deleting regions of the social network that may not be *relevant* to infer the activity of a given target-user. We follow the Smallest Sufficient Region (SSR) approach (Fong and Vedaldi 2017; Dabkowski and Gal 2017), which aims to identify the smallest set of the input that achieves a classification accuracy in line with the general (complete) model. We provide two different solutions by varying the network connectivity of each user and adapting the DNN architecture accordingly.

On the other hand, scalability issues may occur when new users register to the social network. In such a case, G-SIDL requires to be retrained to include the new users in the model. This process is computationally expensive, both in time and resources. Our objective here is to mitigate this issue in the most efficient way while preserving the performance of the G-SIDL approach. We next present the solutions proposed to overcome both these issues.

Local-SIDL (L-SIDL)

Inspired by the SSR approach, we consider the smallest social network within each individual is embedded into. Thereby, we take into account only the set of one-hop neighbors that each user is connected to. The resulting *ego network* is used as the input of each user-based model. We refer to this solution as *Local-SIDL* (L-SIDL), as we create a *local* model for each target-user by employing a DNN for each one of them.

Such a solution can represent a more interpretable and scalable solution if compared to G-SIDL. While in the general approach the post-hoc explanation of the results was blurred by the huge amount of inputs, we here restrict our analysis to the nodes directly connected to the target-user. In such a way, we can better understand whether and to what extent this subset of nodes influence the target user. In terms of scalability, the user-based model offers a more agile solution in terms of time and resource consumption. In fact, a new user in the OSN requires only (*i*) to create a new instance of L-SIDL to model the new user, and (*ii*) to retrain only the L-SIDLs related to the individuals linked to the new user to include her in their model, which both represent operations computationally less expensive (in terms of computational time and resource) than the retraining of the whole G-SIDL. This assessment will be detailed and discussed in "Evaluation" section.

As an example, Fig. 4 shows the L-SIDL related to the test-user u_5, whose ego network is shown in Fig. 1[1]. Each input of the DNN represents a friend of target-user u_5. For each action a, input nodes can assume value 1 if the corresponding friend performed action a before u_5, and 0 otherwise. The output of the DNN $y_{u_5,a}$ corresponds to target-user u_5, and is equal to 1 if she performed a, and is 0 otherwise. The training is performed by minimizing the binary cross-entropy loss between $\hat{y}_{u_5,a}$ and $y_{u_5,a}$, where $\hat{y}_{u_5,a} = f((u_1, u_2, u_3, u_4))|\Theta)$ is the predicted output of the L-SIDL framework. Finally, we utilized the trained model to predict whether the target-user will perform new activities based only on her active friends S_{a,u_5} for those activities.

Community-SIDL (C-SIDL)

While L-SIDL can offer a more interpretable and scalable solution, the prediction performance may be affected by the reduced amount of information that each DNN utilizes. In fact, by splitting the social network into different and *isolated* ego networks, we break the G-SIDL into N L-SIDL models, and in turn, we do not exploit and model the interconnections between the ego-networks. Also, the L-SIDL approach does not take into account nodes distant more than one hop (e.g., friends of a friend) from the target-user. Thus, it assumes that the possible *influencer* nodes are only in the ego network of the target-user.

In this section, we propose *Community-SIDL* (C-SIDL), an approach that aims to solve the trade-off between performance, interpretability, and scalability by embedding users

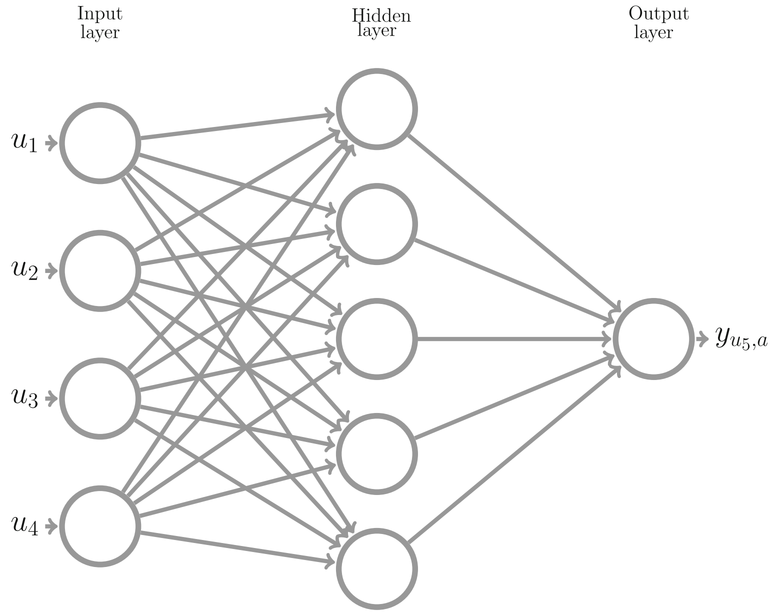


Fig. 4 Local SIDL (L-SIDL) of user u_5, where each input node represents u_5's friend, while the output $y_{u_5,a}$ corresponds to target-user u_5, and is equal to 1 if she performed a, and is 0 otherwise

in mesoscale level structures, such as communities, and employing a DNN architecture for each group of individuals. Such an approach, which can be viewed as a combination of deep learning with network science, offers a solution in-between G-SIDL and L-SIDL. We move from the totality of the nodes (in G-SIDL) to a smaller subset (but larger than a ego-network) by splitting the social network into $1 \leq M \leq N$ communities and by employing a DNN for each of them. We refer to this solution as C-SIDL since each community is mapped into a distinct DNN. The rationale of this approach is to preserve the social network structure at a lower resolution, as a community can be considered as a partition of a graph (Fortunato 2010), while enhancing the interpretability and scalability issues. In fact, by deploying M DNNs, instead of one (G-SIDL), we contain the inefficiency of the G-SIDL in case of new users. In such a scenario, a new user in the OSN requires only to retrain the C-SIDL she belongs to. We aim to partition the network by maximizing the modularity, i.e., the density of intra-community edges with respect to inter-community edges. As this is a NP-hard problem, we make use of a heuristic algorithm. We employ the Louvain method (Blondel et al. 2008) for its computational efficiency [2].

The C-SIDL approach opens the way to multiple solutions at varying inter-community connectivity. Figure 5a shows an example of three connected communities, referred to as C_1, C_2, and C_3. Each community can be considered as an independent component of

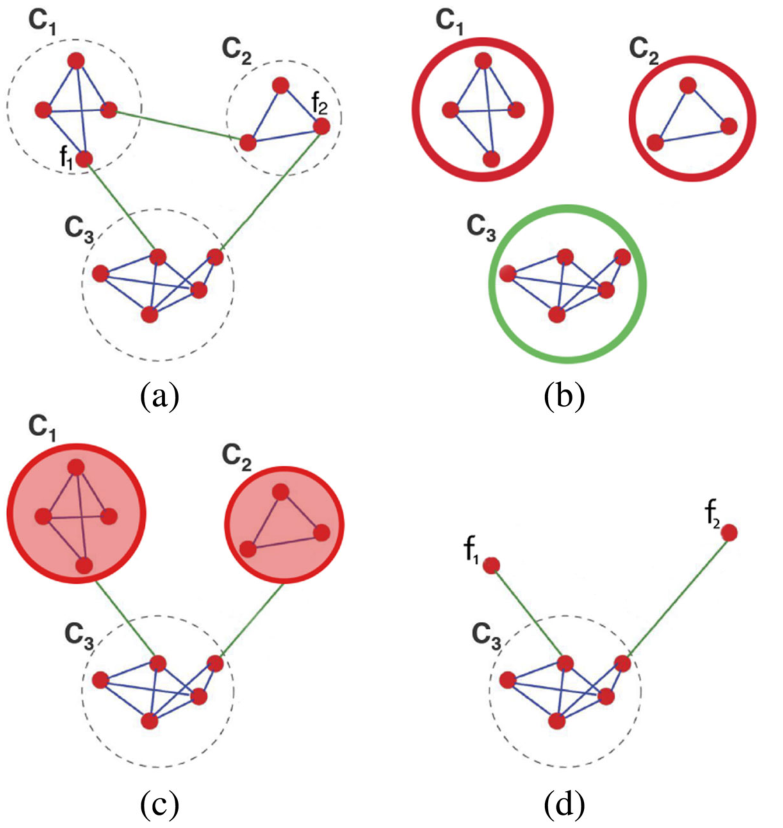


Fig. 5 Example of communities and inter-community connectivity: (**a**) three connected communities; (**b**) IC-SIDL scenario for community C_3; (**c**) CC-SIDL scenario for community C_3; (**d**) IFC-SIDL scenario for community C_3

the graph or as a unit connected with other (linked) communities. In Fig. 5, we show the varying inter-community connectivity of C_3 according to the following approaches:

1. *Isolated Communities-SIDL (IC-SIDL)*: each community is analyzed separately. As shown in Fig. 5b, IC-SIDL does not consider inter-community edges of community C_3. The resulting IC-SIDL architecture is similar to G-SIDL, but the inputs are the only nodes belonging to C_3.
2. *Collapsed Communities-SIDL (CC-SIDL)*: every connected community is considered as a super node linked to the community C_3, as it is depicted in Fig. 5c. In this solution, we collapse an entire community into a single input node (*community node*) of the DNN. The resulting CC-SIDL follows the IC-SIDL implementation but includes also the community nodes as input. As an example, community node C_1 (same for C_2) acts as an input of the CC-SIDL of C_3.
3. *Inter-Friendship Communities-SIDL (IFC-SIDL)*: only users directly connected to C_3 are considered as inputs of the IFC-SIDL, other than the members within C_3 (as in IC-SIDL). Figure 5d shows an example of two users (f_1, f_2) connected to C_3.

Data processing and implementation

In this section, we describe the data processing, we discuss the training, validation, and testing phases of SIDL, and we detail the implementation used for every DNN framework previously described.

Each dataset includes information about users activity over time and social connections among subjects in the OSN, while no additional information or metadata are provided. These data allow us to build the action log $\mathcal{A}_l$ and the graph G, which represent the only two inputs required by our framework. In fact, the former is used to keep track of users activity over time and, in combination with the latter, to understand how the actions propagated between the users. To reduce noise in the dataset and to build, for each user, a reasonable training and test set, we remove users with less than 10 actions. For Plancast, we restrict our analysis to the U.S. as the majority of the users attended events organized in this country.

For each subject u_i, we consider the set of actions the user performed ($A_{u_i} \subseteq A$) and we randomly select n_{u_i} actions not performed in order to consider negative samples, where $n_{u_i} = |A_{u_i}|$. It should be noted that, for each user, we consider only the actions that

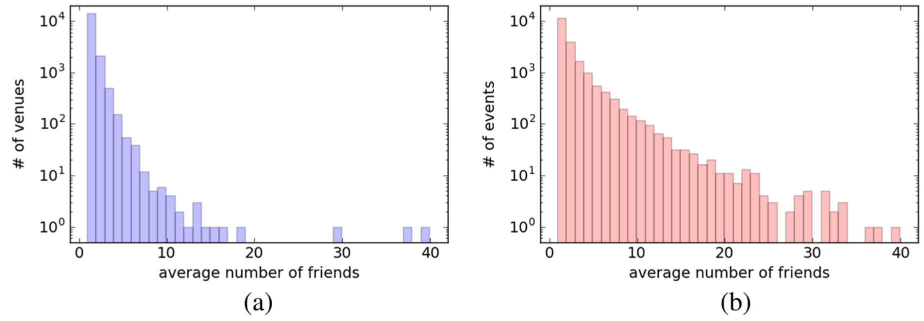


Fig. 6 Average number of friends that visited a given venue (**a**), and that attended a certain event (**b**)

have been performed by at least one of her friends, according to (Goyal et al. 2010). In Fig. 6, we display the average number of friends that visited (attended) a given venue (event). As we expected, events involve more friends, and in turn represent a more direct and expansive form of social interaction, if compared to the visit of a certain location, which may represent mainly individual behaviors (Liu et al. 2012). However, the number of friends that visited a given venue is not negligible and, thus, we consider this scenario in our study.

To limit overfitting and to reduce variability, we utilize a 10-fold cross validation. We built the folds to preserve the percentage of positive and negative samples for each subject in the dataset. Each sample (both for training and test) represents an action performed (or not) by a given target-user. For each action in the training set, we provide information about the target-user's friends activity along with the ground truth related to the target-user activity. On the other hand, for the actions in the test set, we employ only the information related to the target-user's friends activity with the purpose of inferring whether the target-user performed the action. Each friend is mapped into an input node of the DNN, which in turn is a binary value representing the friend activity for a given action. Inputs are fed into the DNN architecture according to the different SIDL approach, as explained in "Methodology" section.

While the input preparation for G-SIDL and L-SIDL is similar, some additional clarifications are needed for the C-SIDL approaches. In C-SIDL every community is mapped into a different DNN. The different C-SIDL approaches differ from each other in the way the inter-community links are considered. As an example, Fig. 5 shows the three different scenarios for community C_3. IC-SIDL, which does not take into account the inter-community links, follows exactly the G-SIDL implementation but its inputs are the only nodes belonging to the community under exam (C_3 in the example in Fig. 5b). On the other side, CC-SIDL and IFC-SIDL consider the inter-community links in two different ways.

In CC-SIDL, each connected community to C_3 is collapsed into a single input node, named *community node*. The resulting DNN complies with the IC-SIDL implementation for the nodes belonging to C_3, but also includes the *community nodes* C_1 and C_2. Each community node is employed as an additional member of C_3 and is considered as a friend of the users directly connected (the top right users in the example in Fig. 5c) to the community node itself. To represent the community node's activity for a given action we tested different strategies. We set the community node to (*i*) a binary value, which assumes value 1 if at least one user within the collapsed community performed the action, (*ii*) a binary value, which assumes value 1 if at least 50% of the users within the collapsed community performed the action, or to (*iii*) a real value representing the fraction of the users within the collapsed community that performed the action. Among the three strategies, we used the first approach as it achieved slightly better results (details on the performance in the next section). Finally, IFC-SIDL considers the inter-community friends as additional members of the community under examination. In the example in Fig. 5d, users f_1 and f_2 are included in the DNN related to community C_3 and considered as friends of the connected nodes.

We implemented the DNNs in Keras (Chollet 2017), following a tower pattern composed of $L = 3$ layers with {128,64,32} nodes, respectively. For the Foursquare dataset, we used an additional layer of 256 neurons as it improved the performance with respect to the three layers architecture used for the Plancast dataset. We tuned hyperparameters performing a grid search on a validation set (10% of the data). In particular, we examined the following hyperparameters:

- *Batch size* defines the number of samples used to update the model at each iteration;
- *Initialization* of the inter-layer weights;
- *Epochs* represent the number of times each sample in the dataset is considered during the training;
- *Optimizer* indicates the optimization algorithm used to update the network weights;
- *Activation function* at the hidden layers ϕ_j and at the output layer ϕ_o are the non-linear function used to *activate* the neurons over all the network.

The hyperparameter optimization consists of an exhaustive searching through the following hyperparameters space: batche size = {10,20}, initialization = {normal, uniform, Glorot}, Epochs = {10,25}, optimizer = {RMSProp, Adam}, activation functions ϕ_j = {sigmoid, ReLU}, and ϕ_o = {sigmoid, ReLU}. Table 2 depicts the six best results, in terms of accuracy, for the combinations of the above hyperparameters. Thereby, we employ a sigmoid as activation function (both at the hidden layers and at the output layer), and we use RMSProp (Dauphin et al. 2015) as optimization function. We train the network in data batches composed of 20 samples for 25 epochs. We further evaluated the impact of the number of epochs in the performance by testing the model for 50 and 100 epochs. Despite the gain of about 0.1% (in the case of 100 epochs), we decided to use 25 epochs for a time efficiency reason, as a larger number of epochs implies a longer training time. Finally, we apply a dropout technique (Srivastava et al. 2014), with a dropout equal to 0.1, to avoid overfitting.

Evaluation

In this Section, we evaluate the results of the SIDL approaches and we compare them with two state of the art approaches, namely the LT model proposed by Goyal et al. (2010), and the IC model of Saito et al. (2008). We use as baseline these two solutions as they offer general models to learn social influence between users by leveraging only the history of the actions performed by each subject. In fact, (*i*) they do not rely either on specific hand-crafted features or on topic affinity, which in turn may depend on the OSN analyzed and on the availability of metadata (e.g., personal attributes), and (*ii*) they both take as input only the action log $\mathcal{A}_l$ and the social graph G. In a similar way, we focus on a model that can be generalized to any kind of OSN and, less specifically, to any real-life domain. Both these approaches rely on two commonly used models in information diffusion, namely the LT and IC model, which we introduced in "Introduction" section. In

Table 2 Grid search for hyperparameters optimization

Accuracy	Batch size	Initialization	Epochs	Optimizer	ϕ_j	ϕ_o
88.7	20	Glorot	25	RMSProp	sigmoid	sigmoid
88.6	10	Glorot	10	RMSProp	sigmoid	sigmoid
88.3	20	normal	10	RMSProp	ReLU	sigmoid
87.8	10	Glorot	10	RMSProp	ReLU	sigmoid
87.7	10	Glorot	10	Adam	sigmoid	sigmoid
85.7	10	Glorot	10	Adam	ReLU	sigmoid

particular, they first aim at learning influence probability between users, and then they combine the social influence model with the diffusion model. In Goyal et al. (2010), Goyal et al. introduced different metrics to estimate the pairwise influence between two individuals and proposed a static and dynamic (time-dependent) version of the LT model. We evaluated all the metrics and variations of the LT model proposed in (Goyal et al. 2010) and we report the results related to the Discrete Time (DT)-Bernoulli approach as it achieved better performance if compared to the other metrics. We refer to this solution as LT-DT indicating the discrete time version of the LT model. On the other hand, Saito et al. (2008) employed the IC model along with the Expectation-Maximization (EM) algorithm to estimate the influence probability associated with each edge. We developed this model, here referred to as IC-EM, by minutely following the 2-steps learning and the experimental setup suggested in their paper (Saito et al. 2008).

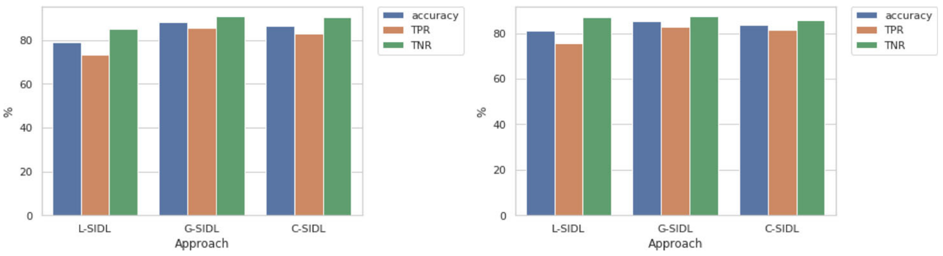
In Table 3, we compare the performance of these solutions with our proposed approaches in terms of prediction *accuracy*. This metric stands for the number of correctly classified samples over all the samples classified. Results indicate that the proposed SIDL approaches, G-SIDL, L-SIDL, and C-SIDL outperform baseline algorithms (LT-DT and IC-EM) with an average gain of 23.7%, 14.4%, and 21.7%, respectively. As we expected, G-SIDL outperforms the local approach, while C-SIDL, with its three variations (IC-, CC-, and IFC-SIDL), offers a valuable alternative to the global solution. Three aspects are worth noting:

- IC-SIDL performs better than L-SIDL as it considers the community within the user is socially embedded, and not only the direct social connections. However, this solution breaks the connectivity between linked communities, thus, performs worse if compared to CC-SIDL and IFC-SIDL.
- CC-SIDL slightly overcomes the IC-SIDL accuracy but has a small gain with respect to L-SIDL. Collapsing an entire community in a unique node oversimplifies the inter-communities social relationships, but provides additional information if compared to the IC-SIDL solution.
- IFC-SIDL achieves the best performance among the C-SIDL approaches and its accuracy closely approaches G-SIDL, highlighting the importance of inter-community edges in modeling social influence.

To better investigate the performance of our approaches, we examine other binary classification metrics, such as True Positive Rate (TPR) and True Negative Rate (TNR). TPR measures the percentage of positive samples that are correctly identified as such, while

Table 3 Models performance in terms of accuracy: G-SIDL vs. L-SIDL vs. C-SIDL (IC-, CC-, and IFC-SIDL) vs. baseline models (LT-DT (Goyal et al. 2010) and IC-EM (Saito et al. 2008))

	Foursquare		Plancast
	NYC	LA	
G-SIDL	89.3%	87.0%	85.1%
L-SIDL	80.1%	78.0%	81.2%
IC-SIDL	84.3%	81.1%	81.8%
CC-SIDL	85.3%	82.2%	82.4%
IFC-SIDL	88.1%	85.3%	83.7%
LT-DT (Goyal et al. 2010)	62.3%	62.0%	77.8%
IC-EM (Saito et al. 2008)	63.2%	62.9%	76.9%



(a) (b)

Fig. 7 Accuracy, TPR, and TNR in Foursquare (**a**), and Plancast (**b**)

TNR is the analog for negative samples, i.e., it measures the percentage of correctly classified negative instances. In our scenario, these two metrics represent the ability of the classifier to identify performed action and not performed action, respectively. In Fig. 7, we compare our approaches using accuracy, TPR, and TNR. Please note that for the C-SIDL approach we consider IFC-SIDL as it outperforms the other two C-SIDL solutions (IC-SIDL and CC-SIDL). Overall, we observe that TNR is higher than TPR in every approach, meaning that our system (slightly) better classifies negative samples. The difference between TPR and TNR is more pronounced in the L-SIDL approach, while in C-SIDL and, especially, in G-SIDL the two metrics are more balanced. Interestingly, the difference between TPR and TSR is less noticeable in Plancast than in Foursquare, probably because the number of friends per visited location is significantly lower if compared to Plancast events, as we previously showed in Fig. 6, and the model, in turn, is less accurate to classify this kind of activity.

In Table 4 we summarize the performance of the three SIDL approaches and compare them in terms of scalability. We sort the table in increasing order of computational time or decreasing order of number of DNNs employed. Computational time is the time required to train a single L-SIDL, C-SIDL, and the unique G-SIDL. Although the G-SIDL approach achieves the best performance, it requires a (unique) DNN with an elevated number of inputs (two times the number of nodes in the social network), which is not scalable for a huge graph. In fact, retraining G-SIDL every time a new user register to the OSN is time and resource consuming. On the other side, L-SIDL offers more flexibility and efficiency as every user is modeled with a different DNN. Therefore, this approach may easily handle new users in the OSN by creating a DNN for the new user and updating only the DNNs of the new user's friends. The input size of each L-SIDL equals the number of friends of the target-user. Thus, the corresponding DNN is significantly smaller than the global neural network (G-SIDL). For this reason, the training phase of a single L-SIDL

Table 4 Comparison among the three presented approaches in terms of performance and scalability

	L-SIDL	C-SIDL	G-SIDL
Accuracy	80.1%	85.2%	86.6%
TPR	74.4%	82.4%	84.3%
TNR	85.9%	88.1%	88.7%
Computational time	t	7.8t	75.1t
# of DNNs	N	L	1

Computational time is the time required to train a single L-SIDL, C-SIDL, and the unique G-SIDL. The value *t* is about 42 s, which has been computed by averaging the computational time of a single L-SIDL over the datasets is, on average, 75 times faster than the G-SIDL. However, the computational efficiency of the L-SIDL approach is paid in terms of performance. The local solution breaks the whole social network structure in disconnected ego networks and, thus, performs poorly (average accuracy of 80.1%) if compared to G-SIDL (average accuracy of 86.6%).

The trade-off is solved by C-SIDL, which achieves performance close to the G-SIDL with limited issues in scalability: a new user in the OSN requires only to retrain one C-SIDL, whose computational time is about 10 times faster than G-SIDL and 8 times slower compared to L-SIDL[3]. Note that the number M of communities, and in turn the number of DNNs in C-SIDL, depends on the connections between users in the social network and on the detection algorithm utilized to extract the communities. As a consequence, the input size of a C-SIDL depends on the number of members per each community. Further, we investigate whether the number of members per community has impacted on the prediction performance. For this purpose, we compute the Pearson correlation coefficient between the accuracy and the number of members per community. The result shows a not statistically significant correlation (ρ=0.3, p-value=0.17 $>$ α=0.05). We then explore whether and to what extent connected communities share their activities. To accomplish this purpose, for each community, we compare the actions performed by its members, namely *intra-community actions*, with those performed by the members of the inter-connected communities, referred to as *inter-community actions*. More specifically, for each community, we compute the fraction of users that performed a given activity considering (i) only the members of the community, and (ii) considering both the members of the community and of the inter-connected communities. Figure 8 depicts the results for both datasets. As we expected intra-community actions present higher fractions of users if compared to the combined (intra- with inter-community) scenario. However, the contribution provided by the inter-connected communities is not negligible and, according to the prediction performance, plays a significant role. Interestingly, in Plancast both the percentage of intra- and inter-community actions is higher with respect to Foursquare, further highlighting the differences we revealed before. In the Plancast dataset, we can also note that for some activities every member of the community performed the action. After further inspection, we recognize that these activities correspond to small clique events, which involved communities composed of a few members.

Finally, the results of C-SIDL and L-SIDL give us a better understanding in the post-hoc explanation of the results obtained with G-SIDL. Though the one-hop neighborhood (ego network) appeared as the straightforward solution to detect the set of input (nodes)

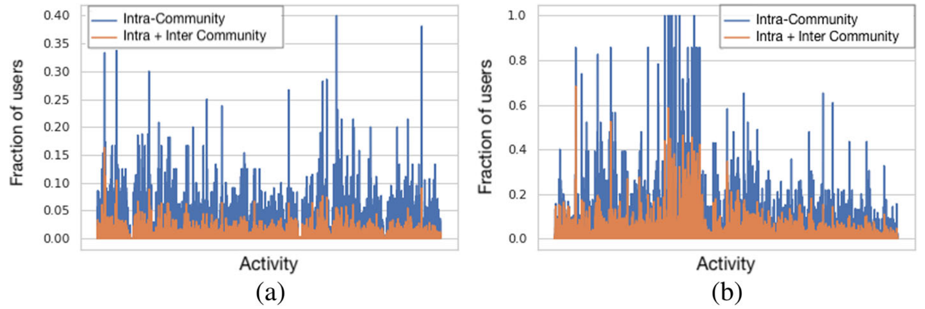


Fig. 8 Intra- and Inter-community activity in Foursquare (**a**), and Plancast (**b**)

that affected the prediction related to a given user, results show that the ego network alone is not enough to explain the social influence phenomenon. As we discussed in Luceri et al. (2017), mesolevel structures, such as communities, provide a significant contribution to the understanding of this phenomenon. Results show that C-SIDL closely approaches the performance of the general model. Therefore, the SSR identified by a community better explains the results achieved by G-SIDL if compared to the SSR built with the ego network only. In these terms, C-SIDL provides better interpretability of the global model if compared to L-SIDL. Overall, we can further explain the gap between L-SIDL and C-SIDL in terms of influential nodes considered within each social structure. The rationale of L-SIDL is that the most influential nodes stand in the ego network of each user, while in C-SIDL we enlarge the social structure within the user is embedded to include (additional) potential influencers not (only) directly connected to the target-user.

Discussion and future directions

In this Section, we discuss further research questions and potential extensions of this work along with some preliminary results. In the first place, this paper opens the way to necessary consideration and discussion on users' privacy in OSNs, and in particular, on the relation between social influence and privacy. Results revealed that our approach is able to estimate an individual's actions with high accuracy based only on the knowledge about friends' activity. User activity and behavior, in turn, can represent sensible information, which a subject may not be willing to share in some (or future) instances. However, recent research showed that the behavior of individuals is predictable using only the information provided by their friends in an online social network (Luceri et al. 2019; Bagrow et al. 2019; Garcia 2019). Interestingly, Bagrow et al. (2019) reported that friends information in Twitter can be used as a proxy to predict future behavior of an individual at a higher degree than the data shared from the subject herself.

To further investigate this argument, we repeat our experiments in a different setting. We evaluate the prediction accuracy of SIDL by varying the probability p that each individual's friend shares the information about a given activity, i.e., if a friend performed a certain activity, we exploit that knowledge with probability p. In this analysis, we use G-SIDL as we are mainly interested in exploring the privacy lack in OSNs rather than

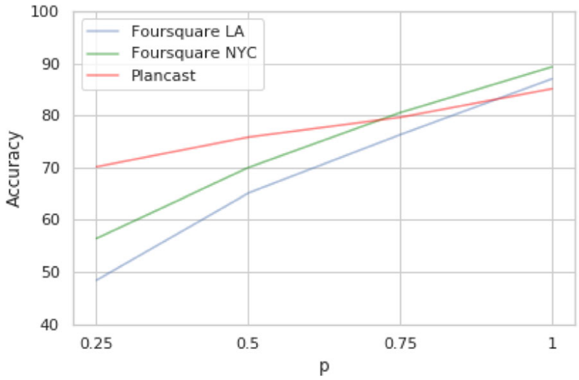


Fig. 9 Prediction Accuracy at varying sharing probability p

focusing on the interpretability or scalability of the model. In Fig. 9, we depict the accuracy of our model at varying p. Diverse aspects are worth of consideration. There is a noticeable gap between the performance in Plancast and Foursquare. This is likely due to higher involvement of friends in a social event compared to a visit in a certain location, as we previously showed. Further, we can observe that our model, on average, is able to correctly classify about 70% of users' activities in case of $p = 0.5$. This result further highlights the weakness of data privacy in social platforms: with only 50% of available information from a user's friends we are able to classify around 70% of her activities. Such findings suggest that users' information domain is not only confined to what they deliberately share. Privacy is not only in users' hands as friends and acquaintances act as social signals, which can be powerfully used to estimate users' sensitive data. The concern about data privacy is tangible and our society is entirely involved with the issue of privacy leakage in OSNs. As Garcia (2019) claims, we need to stop thinking that the decision to keep information private is under individual control and realize that information secrecy may be affected by the decisions of others. In accordance with this discussion, in our next work, we will expand this analysis following two main directions. First, we have been carrying out a campaign to raise user awareness of privacy risks when using mobile phones (Ferrari and Giordano 2018; Luceri et al. 2018). The idea is to examine whether users reduce the amount of shared information according to an increasing level of privacy awareness and, at the same time, to observe if the same effect can be observed in an OSN context. Second, we will focus on developing a model that provides an assessment of individuals' privacy, given both the information they voluntarily disclose and the data provided by their friends over time.

Moreover, we explore the usage of different deep learning architectures across our framework. In the current version, SIDL uses feedforward neural networks, a class of DNNs in which the flow of information moves *forward* from the input to the output neurons through the hidden layers. In such architecture, there are no loops or cycles between the nodes of the network. However, feedback connections have been extensively used in *Recurrent* Neural Networks (RNNs) for modeling the temporal dynamics of sequences in a wide range of fields, such as machine translation, speech, handwriting, and image recognition. The feedback loop allows RNNs to have memory of previous instances and makes them suitable to learn sequential data, such as human activities. For this reason, we propose to use a RNN architecture in the SIDL approach. We expect that such a solution might be more beneficial in modeling users actions and dependencies over time with respect to a feedforward architecture. The most commonly used RNNs are referred to as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (GRU) (Chung et al. 2014). We do not present further details about these solutions as it is out of the scope of this paper. However, we here show some preliminary results obtained using both the RNN approaches in our framework. More specifically, in

Table 5 Architecture performance in terms of accuracy: Feedforward vs. LSTM vs. GRU

	Foursquare		Plancast
	NYC	LA	
Feedforward	89.3%	87.0%	85.1%
LSTM	91.1%	89.2%	85.6%
GRU	86.7%	85.3%	83.3%

this test, we consider G-SIDL as it achieves the best prediction performance, but the same architecture can be extended to the other approaches. In Table 5, we compare the results of the three DNN architectures on the two datasets. We observe that LSTM outperforms the GRU solution and achieves a better accuracy of the feedforward architecture. Interestingly, the gap is particularly noticeable in the Foursquare dataset. This may be due to the different nature of the activity in the two datasets. While an event is a one-shot activity held on a specific date, the visit of a certain location may occur in distinct days from one user to another. This hypothesis, with related analysis and a broader exploration of DNN architectures, will be expanded upon in future work.

Finally, in our next endeavor, we aim to investigate other factors that may affect social influence to better comprehend this phenomenon and to further enhance the predictive power of our approach. We can identify two parallel directions. In the first one, we will focus on the concept of homophily (McPherson et al. 2001) with the objective of considering the similarity between users in our model. In the second direction, we will evaluate different solutions to group users in mesolevel structures and employ C-SIDL accordingly. A wide spectrum of possibilities can be explored for this purpose, ranging from homophily community (Luceri et al. 2017), which groups together similar individuals with low social distance (Förster et al. 2012), to tensor decomposition techniques (Kolda and Bader 2009), which extract sub-network modules composed of nodes with correlated activity.

Conclusion

In this paper, we presented SIDL, a framework that combines deep learning with network science for modeling and forecasting social influence on real-life activities. To the best of our knowledge, this is the first work that aims to provide a social influence model of real-world activities.

SIDL is based on DNNs, a class of machine learning algorithms inspired by biological nervous systems. We propose different SIDL approaches at varying degree of network connectivity with the objective of facing two typical challenges of deep neural networks: interpretability and scalability. For this purpose, we introduce G-SIDL, which takes into account the whole social network; L-SIDL, which considers the ego network of each user; and finally C-SIDL, which splits G-SIDL into smaller sub-models based on social network communities.

We validate and evaluate our approach using datasets from Plancast and Foursquare. Results reveal that SIDL approaches outperform state of the art baselines with an average gain of 23.7% (G-SIDL), 14.4% (L-SIDL), and 21.7% (C-SIDL), respectively. We show that the opportune combination of network science with deep learning can address both the interpretability and scalability issue. In fact, C-SIDL provides a post-hoc explanation of the results by identifying a subset of users, which are mainly responsible for the prediction outcome, using social network mesolevel structures. Further, C-SIDL closely approaches the performance accuracy of the global approach, also providing a more scalable model.

Moreover, we explore the possibility of using different deep learning architectures across our framework by showing some promising results. Finally, we discuss the relationship between social influence and users' privacy presenting alarming findings: with only 50% of available information from a user's friends SIDL is able to classify around 70%

of user's activities. These two directions, along with an analysis on the homophily effect and on alternative approaches to detect mesolevel structures, will be expanded in our future work.

Endnotes

[1] Also in this Figure, for the sake of simplicity, we depict a DNN with only one hidden layer ($L = 1$), while in our implementation ("Data processing and implementation" section) we employ multiple hidden layers.

[2] As a sanity check, we repeated the community extraction 10 times and we evaluated the difference in the partitions utilizing the Normalized Mutual Information (NMI). In every round, the Louvain method extracted the same number of communities, which in turn did not vary across the different rounds, i.e., the NMI was equal to one for each pair of rounds.

[3] It should be noticed that computational times are averaged over the different datasets. We run our experiment on a machine with a NVIDIA Tesla K20 (2496 CUDA cores - 5GB DDR5 RAM), a CPU Intel Xeon E5 2670 with a frequency of 2.3 GHz, and a 128 GB DDR3 RAM.

Abbreviations

C-SIDL: Community-SIDL; DNN: Deep neural network; EBSN: Event-based social network; EM: Expectation-maximization; eWOM: Electronic word of mouth; G-SIDL: Global-SIDL; GRU: Gated recurrent unit; IC: Independent cascade; L-SIDL: Local-SIDL; LBSN: Location-based social network; LSTM: Long short-term memory; LT: Linear threshold; NMI: Normalized mutual information; OSN: Online social network; SIDL: Social influence deep learning; SM: Saliency mask; TNR: True negative rate; TPR: True positive rate

Acknowledgements

We are very thankful to our colleagues, Davide Andreoletti, Felipe Cardoso and Alan Ferrari, for very inspiring discussions; and to the anonymous reviewers for their invaluable comments.

Funding

This work is funded by the Swiss National Science Foundation (SNSF) via the CHIST-ERA project *UPRISE-IoT*.

Availability of data and materials

Please contact author for data requests.

Authors' contributions

LL carried out the study, conceived the framework, and drafted the manuscript. SG participated in the design of the study, in the conception of the framework, and in finalizing the manuscript. TB participated in the analysis of the results, and helped to draft the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not Applicable.

Consent for publication

Not Applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 7 November 2018 Accepted: 16 April 2019
Published online: 13 June 2019

References

Albert R, Barabási A-L (2002) Statistical mechanics of complex networks. Rev Mod Phys 74(1):47

Anagnostopoulos A, Kumar R, Mahdian M (2008) Influence and correlation in social networks. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 7–15

Aral S, Muchnik L, Sundararajan A (2009) Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. Proc Natl Acad Sci 106(51):21544–21549

Bagrow JP, Liu X, Mitchell L (2019) Information flow reveals prediction limits in online social activity. Nat Hum Behav 3(2):122. Nature Publishing Group

Bakshy E, Hofman JM, Mason WA, Watts DJ (2011) Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. ACM. pp 65–74

Bao J, Zheng Y, Mokbel MF (2012) Location-based and preference-aware recommendation using sparse geo-social networking data. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems. ACM. pp 199–208

Bessi A, Coletto M, Davidescu GA, Scala A, Caldarelli G, Quattrociocchi W (2015) Science vs conspiracy: Collective narratives in the age of misinformation. PLoS ONE 10(2):0118093

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008. IOP Publishing

Brown J, Broderick AJ, Lee N (2007) Word of mouth communication within online communities: Conceptualizing the online social network. J Interact Mark 21(3):2–20

Castellano C, Fortunato S, Loreto V (2009) Statistical physics of social dynamics. Rev Mod Phys 81(2):591

Chollet F (2017) Keras 2015. http://keras.io

Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555

Crandall D, Cosley D, Huttenlocher D, Kleinberg J, Suri S (2008) Feedback effects between similarity and social influence in online communities. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 160–168

Dabkowski P, Gal Y (2017) Real time image saliency for black box classifiers. In: Advances in Neural Information Processing Systems. pp 6967–6976

Dauphin Y, de Vries H, Bengio Y (2015) Equilibrated adaptive learning rates for non-convex optimization. In: Advances in Neural Information Processing Systems. pp 1504–1512

Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 57–66

Esfandyari A, Zignani M, Gaito S, Rossi GP (2016) Impact of offline events on online link creation: a case study on events advertised on facebook. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. ACM. pp 1186–1188

Fang X, Hu PJ-H, Li Z, Tsai W (2013) Predicting adoption probabilities in social networks. Inf Syst Res 24(1):128–145. INFORMS

Ferrara E (2015) Manipulation and abuse on social media by emilio ferrara with ching-man au yeung as coordinator. ACM SIGWEB Newsl Spring:4

Ferrara E, Varol O, Davis C, Menczer F, Flammini A (2016) The rise of social bots. Commun ACM 59(7):96–104

Ferrari A, Giordano S (2018) A study on users' privacy perception with smart devices. arXiv preprint arXiv:1809.00392

Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. arXiv preprint arXiv:1704.03296

Förster A, Garg K, Nguyen H-A, Giordano S (2012) On context awareness and social distance in human mobility traces. In: Proceedings of the third ACM international workshop on Mobile Opportunistic Networks. ACM. pp 5–12

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3-5):75–174

Gao H, Tang J, Liu H (2012) Exploring social-historical ties on location-based social networks. In: Sixth International AAAI Conference on Weblogs and Social Media

Garcia D (2019) Privacy beyond the individual. Nat Hum Behav 3(2):112. Nature Publishing Group

Georgiev P, Noulas A, Mascolo C (2014) The call of the crowd: Event participation in location-based social services. arXiv preprint arXiv:1403.7657

Goldenberg J, Libai B, Muller E (2001) Talk of the network: A complex systems look at the underlying process of word-of-mouth. Mark Lett 12(3):211–223. Springer

Goyal A, Bonchi F, Lakshmanan LV (2010) Learning influence probabilities in social networks. In: Proceedings of the third ACM international conference on Web search and data mining. ACM. pp 241–250

Gruhl D, Guha R, Liben-Nowell D, Tomkins A (2004) Information diffusion through blogspace. In: Proceedings of the 13th international conference on World Wide Web. ACM. pp 491–501

Guille A, Hacid H, Favre C, Zighed DA (2013) Information diffusion in online social networks: A survey. ACM Sigmod Rec 42(2):17–28

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. ACM Comput Surv (CSUR) 51(5):93

He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee. pp 173–182

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Huete-Alcocer N (2017) A literature review of word of mouth and electronic word of mouth: Implications for consumer behavior. Front Psychol 8:1256

Hwang T, Pearce I, Nanis M (2012) Socialbots: Voices from the fronts. Interactions 19(2):38–45

Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 137–146

Kimura M, Saito K, Nakano R (2007) Extracting influential nodes for information diffusion on a social network. In: AAAI Vol. 7. pp 1371–1376

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500

La Fond T, Neville J (2010) Randomization tests for distinguishing social influence and homophily effects. In: Proceedings of the 19th international conference on World wide web. ACM. pp 601–610

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436. Nature Publishing Group

Lipton ZC (2016) The mythos of model interpretability. arXiv preprint arXiv:1606.03490

Liu X, He Q, Tian Y, Lee W-C, McPherson J, Han J (2012) Event-based social networks: linking the online and offline social worlds. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 1032–1040

Liu L, Tang J, Han J, Yang S (2012) Learning influence from heterogeneous social networks. Data Min Knowl Disc 25(3):511–544. Springer

Luceri L (2016) Infer mobility patterns and social dynamics for modelling human behaviour. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). IEEE. pp 1223–1224

Luceri L, Andreoletti D, Giordano S (2019) Infringement of tweets geo-location privacy: an approach based on graph convolutional neural networks. arXiv preprint arXiv:1903.11206

Luceri L, Braun T, Giordano S (2018) Social influence (deep) learning for human behavior prediction. In: International Workshop on Complex Networks. Springer. pp 261–269

Luceri L, Cardoso F, Papandrea M, Giordano S, Buwaya J, Kundig S, Angelopoulos CM, Rolim J, Zhao Z, Carrera JL, et al (2018) Vivo: A secure, privacy-preserving, and real-time crowd-sensing framework for the internet of things. Pervasive Mob Comput 49:126–138. Elsevier

Luceri L, Deb A, Badawy A, Ferrara E (2019) Red bots do it better: Comparative analysis of social bot partisan behavior. In: Companion Proceedings of the 2019 World Wide Web Conference

Luceri L, Vancheri A, Braun T, Giordano S (2017) On the social influence in human behavior: Physical, homophily, and social communities. In: International Conference on Complex Networks and Their Applications. Springer. pp 856–868

Matsubara Y, Sakurai Y, Prakash BA, Li L, Faloutsos C (2012) Rise and fall patterns of information diffusion: model and implications. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 6–14

McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: Homophily in social networks. Annu Rev Sociol 27(1):415–444

Metaxas PT, Mustafaraj E (2012) Social media and the elections. Science 338(6106):472–473

Mønsted B, Sapieżyński P, Ferrara E, Lehmann S (2017) Evidence of complex contagion of information in social media: An experiment using twitter bots. PLoS ONE 12(9):0184148

Myers SA, Zhu C, Leskovec J (2012) Information diffusion and external influence in networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. pp 33–41

Newman ME (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256

Noulas A, Scellato S, Mascolo C, Pontil M (2011) An empirical study of geographic user activity patterns in foursquare. In: Fifth international AAAI conference on weblogs and social media

Ratkiewicz J, Conover M, Meiss MR, Gonçalves B, Flammini A, Menczer F (2011) Detecting and tracking political abuse in social media. ICWSM 11:297–304

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 61–70

Saito K, Nakano R, Kimura M (2008) Prediction of information diffusion probabilities for independent cascade model. In: International conference on knowledge-based and intelligent information and engineering systems. Springer. pp 67–75

Schmidhuber J (2015) Deep learning in neural networks: An overview. Neural Netw 61:85–117. Elsevier

Singla P, Richardson M (2008) Yes, there is a correlation:-from social networks to personal behavior on the web. In: Proceedings of the 17th international conference on World Wide Web. ACM. pp 655–664

Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Subrahmanian V, Azaria A, Durst S, Kagan V, Galstyan A, Lerman K, Zhu L, Ferrara E, Flammini A, Menczer F, et al. (2016) The darpa twitter bot challenge. arXiv preprint arXiv:1601.05140

Tang J, Sun J, Wang C, Yang Z (2009) Social influence analysis in large-scale networks. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 807–816

Tangherlini TR, Roychowdhury V, Glenn B, Crespi CM, Bandari R, Wadia A, Falahi M, Ebrahimzadeh E, Bastani R (2016) "mommy blogs" and the vaccination exemption narrative: results from a machine-learning approach for story aggregation on parenting social media sites. JMIR Public Health Surveill 2(2)

Zhang J, Liu B, Tang J, Chen T, Li J (2013) Social influence locality for modeling retweeting behaviors. In: IJCAI Vol. 13. pp 2761–2767

Zhang J, Tang J, Li J, Liu Y, Xing C (2015) Who influenced you? predicting retweet via social influence locality. ACM Trans Knowl Discov Data (TKDD) 9(3):25